

ALGORITHMS AND SOFTWARE FOR
NONLINEAR STRUCTURAL DYNAMICS

N89-29807

58-39

248

211975

N 668 3851

PRINCIPAL INVESTIGATOR:
Ted Belytschko

RESEARCH ASSISTANTS:
Noreen D. Gilbertsen
Mark O. Neal

Northwestern University
Evanston, IL 60208

NASA Grant NAG-1-650
Contract Monitor: Jerry Housner
March 1, 1986 to May 1, 1988

PRECEDING PAGE BLANK NOT FILMED

OUTLINE

OBJECTIVE

EFFICIENT IMPLEMENTATION OF

MIXED Δt EXPLICIT NONL. STRUCT.

DYN. CODE ON VECTORIZED,

(II) CONCURRENT COMPUTER

TOPICS

MIXED Δt INTEGRATION

CONFLICT IN (II) + VECTORIZATION

ALGORITHM

EXAMPLES

ALGORITHMS AND SOFTWARE FOR NONLINEAR STRUCTURAL DYNAMICS

Ted Belytschko, Noreen D. Gilbertsen, and Mark O. Neal

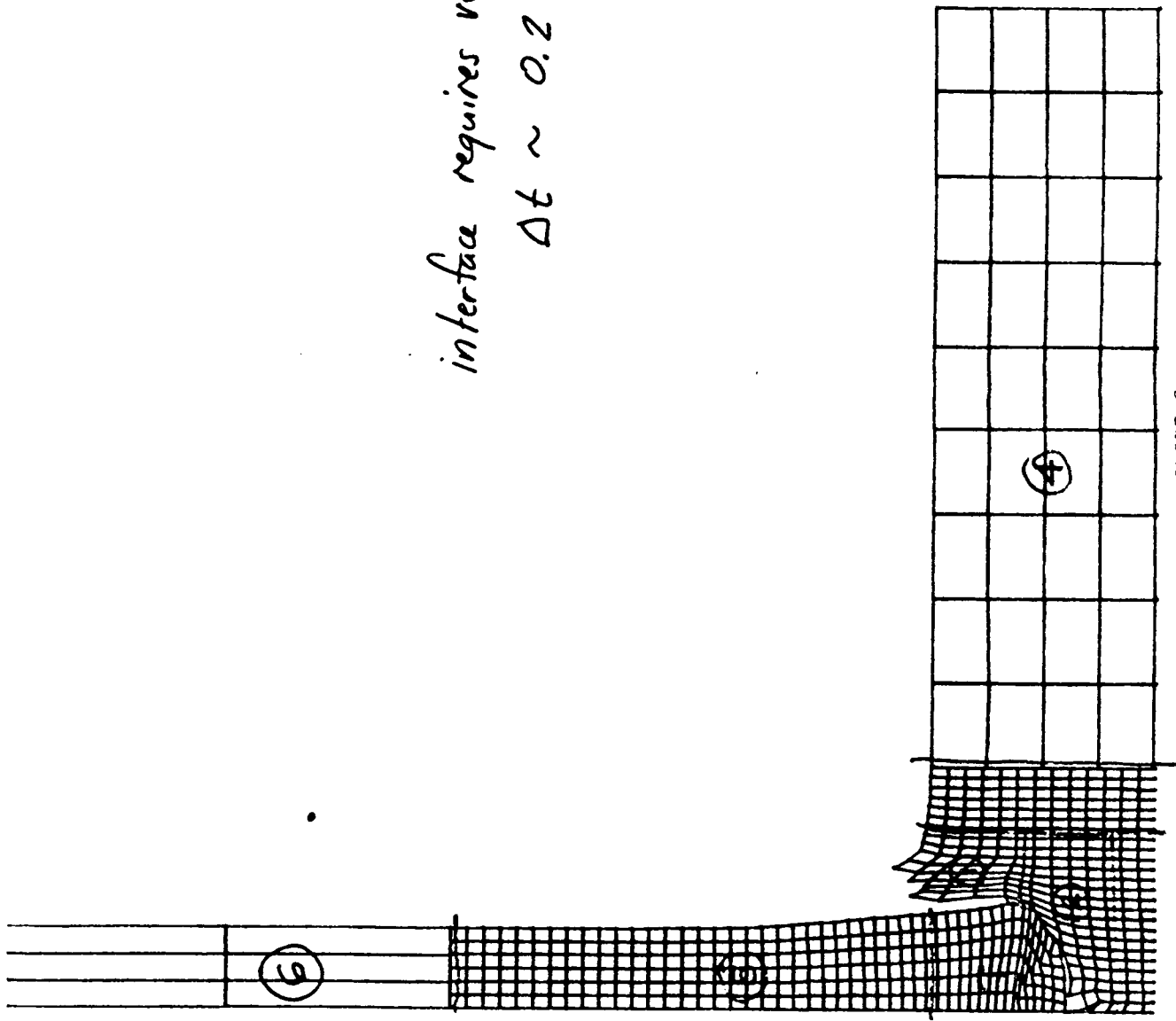
SLIDE 1

The objective of this research is to develop efficient methods for explicit time integration in nonlinear structural dynamics for computers which utilize both concurrency and vectorization. As a framework for these studies, the program WHAMS, which is described in "Explicit Algorithms for the Nonlinear Dynamics of Shells" (T. Belytschko, J. I. Lin, and C.-S. Tsay, Computer Methods in Applied Mechanics and Engineering, Vol. 42, 1984, pp. 225-251), is used.

There are two factors which make the development of efficient concurrent explicit time integration programs a challenge in a structural dynamics program:

1. the need for a variety of element types, which complicates the scheduling-allocation problem;
2. the need for different time steps in different parts of the mesh, which is here called mixed Δt integration, so that a few stiff elements do not reduce the time steps throughout the mesh.

W. H. A.



SLIDE 2

interface requires very small Δt
 $\Delta t \sim 0.2 \Delta t_{crit}$

SLIDE 2

This mesh illustrates why mixed time integration is crucial in the application of explicit methods to structural dynamics. In this mesh, if a single time step were used throughout the mesh, it would be set by the very smallest elements in the mesh and the requirement of the interface. If different time steps are used in different parts of the mesh, a time step in each sub-domain depends on the size of the elements and the requirements of that sub-domain. Thus, much larger time steps can be used in sub-domains 3, 4, 5, and 6.

1st order semidiscretization

$$M \dot{y} + f^{\text{int}} = f^{\text{ext}}$$

for linear $f^{\text{int}} = Ky$

$$\dot{y} = M^{-1} \left(\underbrace{f^{\text{ext}} - f^{\text{int}}}_{f} \right)$$

explicit (Euler forward)

$$y^{n+1} = y^n + \Delta t \dot{y}^n$$

$$= y^n + \Delta t M^{-1} f^n$$

SLIDE 3

This slide shows the fundamental equations involved in an explicit time integration of a first-order system such as the heat conduction equations. The procedures for second-order equations are a little more complicated but fundamentally similar in principle.

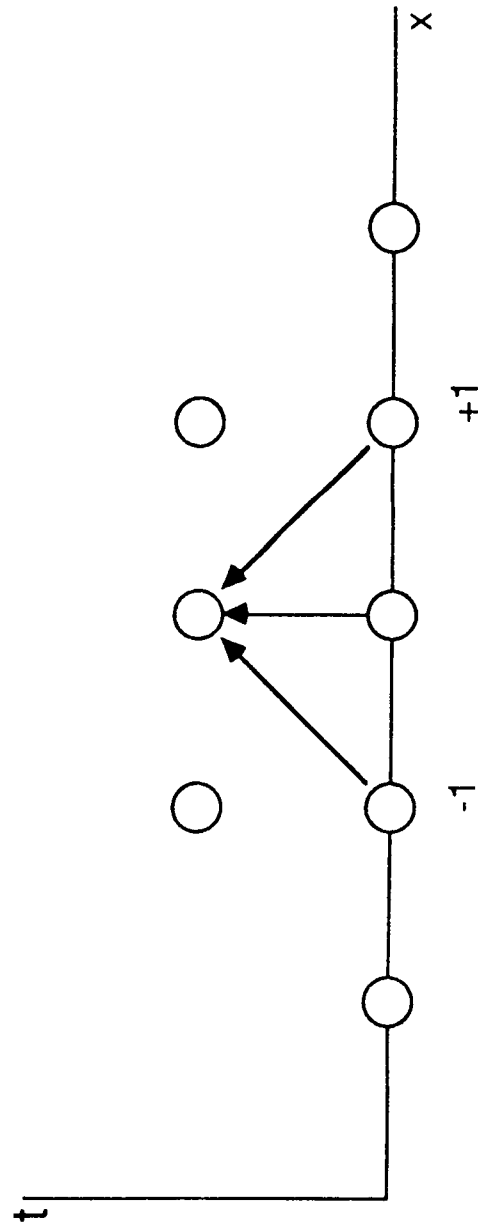
In this slide, \tilde{M} designates the capacitance (mass) matrices, matrix \tilde{f}^{int} the matrix of internal fluxes, and \tilde{f}^{ext} the matrix of external fluxes. As can be seen from the last equation, if the capacitance matrix is diagonal, updating the system from state n to state $n+1$ involves no solution of any equations.

Flow of information

$$u^{n+1} = u^n + M^{-1} (f^{\text{ext}} - K u)$$

in 1D: row of $K \dots$ 0 -1 2 -1 0 \dots

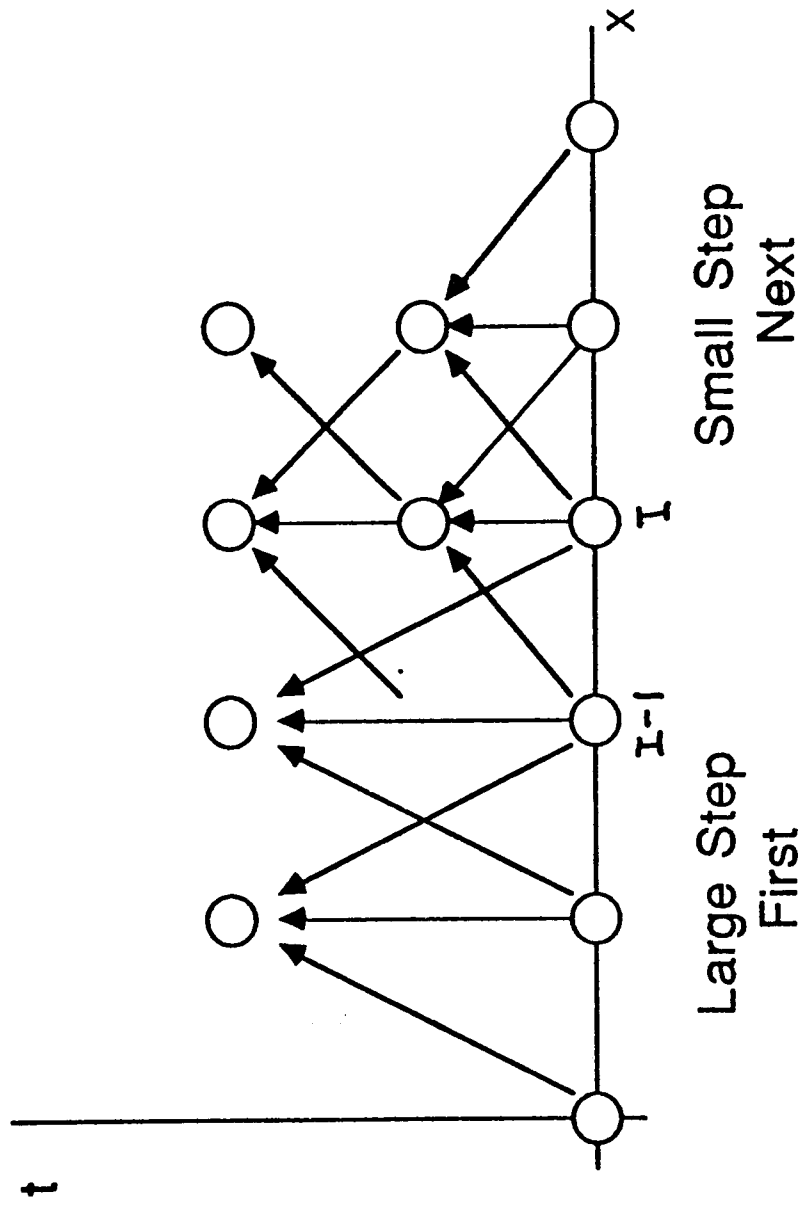
-1 +1



SLIDE 4

In order to examine how a multi-time partition is constructed, it is worthwhile to consider the flow of information in a one-dimensional problem with a single time step for explicit time integration. As can be seen from this figure, the state at time step $n+1$ at any point only depends on the states of the adjacent points at a previous time step. This is represented by the arrows in the figure.

Explicit - explicit nodal partition



SLIDE 5

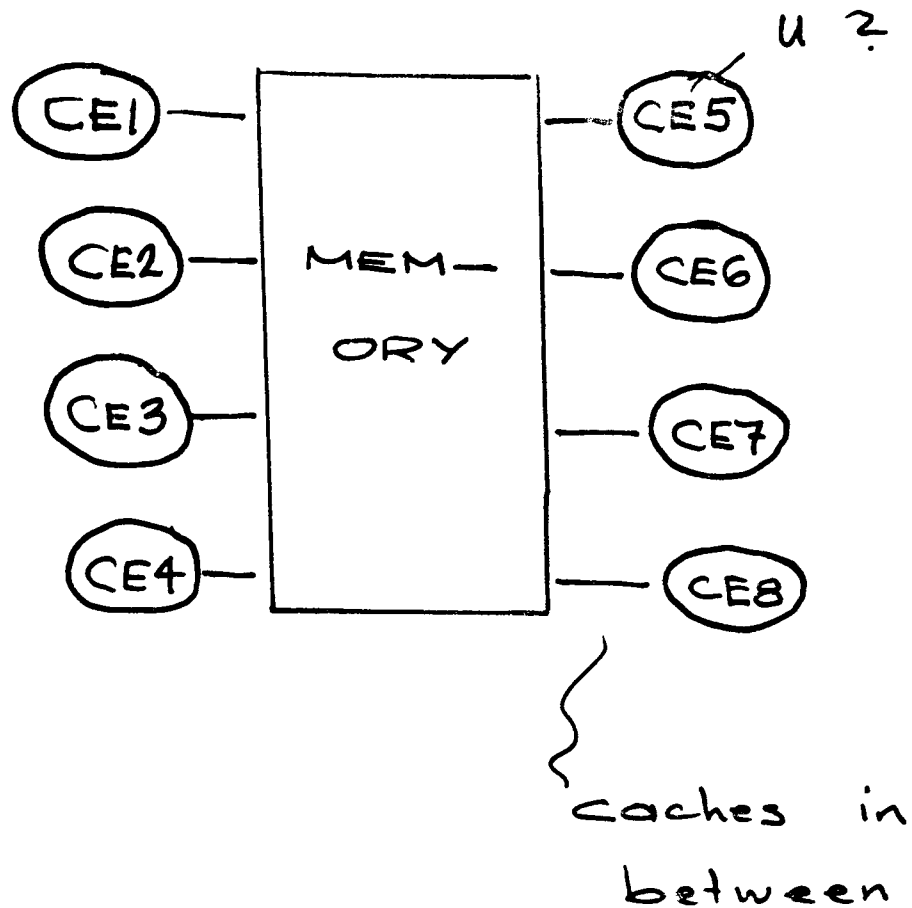
When we consider this flow of information in a one-dimensional problem with a time-step ratio of 2 to 1, then we can see that the standard procedure can be used at all time steps, except at node I which is the small node time step immediately adjacent to the large node time step. In order to update this node, one needs an interpolated value at node I-1, which is indicated by an x in the figure.

The flow of information for the second-order systems which appear in structural dynamics is identical. In both cases, we have used linear interpolations for u at node I-1 in order to get the intermediate values.

These procedures have been used with ratios of time steps as large as 10 to 1 between adjacent elements. The use of these multi-time step partitions does not seem to diminish the stability. For example, if element-type stability criteria are satisfied in both sub-domains for the first-order equations, stability is maintained in the entire system. This has been proven for a slightly different partition for a linear first-order system in "Stability of Multi-Time Step Partitioned Integrators for First-Order Finite Element Systems" (T. Belytschko, P. Smolinski, and W. K. Liu, Computer Methods in Applied Mechanics and Engineering, Vol. 49, 1985, pp. 281-297).

For second-order systems, no proofs of such stability criteria are yet available. However, numerical experiments indicated that the schemes we have used are stable whenever the elements in each sub-domain satisfy a local Courant condition.

ALLIANT FX/8



MIMD (multipl instr. multipl data)
shared-memory computer

- memory contention

cu's are vectorized

SLIDE 6

The computer program has been implemented on an Alliant FX/8, which is an MIMD (multiple instruction multiple data) shared-memory computer. A schematic of the computer architecture is shown. As can be seen, this is a shared memory computer, so it suffers from the problems of memory contention.

A second difficulty is presented by the fact that the computational units on this computer are vectorized. In vectorized computers, finite element computations have to be done on blocks of elements so that vectorization can be exploited. However, in a concurrent machine, the addition of vectorization presents a major dilemma: how long should the blocks be?

If rather long blocks are used, for example, 64 to 128 elements, then a large number of the processors will often be idle. Furthermore, in a concurrent machine, vectorization with long blocks requires considerable additional storage. On the other hand, short element blocks don't take full advantage of vectorization.

One advantage of blocking is that it has provided us with a natural scheme for assigning time steps to groups of elements. In previous implementations of multi-time step integration, the user has had to select the sub-domains. Since, in a vectorized computer, the blocking of elements is already a required task, we have combined this task with the assignment of time steps so that each sub-domain is assigned a time step automatically.

MIXED TIME INTEGRATION ALGRM

1. set initial cond's. : $\underline{u}^0, \dot{\underline{u}}^{-1/2}, n=0, n_{cyc}=0$

2. initialize clocks

$t_{MAST} = 0$ master time

$t_B = 0 \quad B=1, NB$ element block times

$t_N = 0 \quad N=1, N_{node}$ nodal times

3. element block loop

zero \underline{f}^{int} ; for $B=1$ to NB

if $t_B \leq t_{MAST}$

update stress, \underline{I}^{n+1} for $e \in B$

update $\underline{f}_e^{int} = \int \underline{B}^T \underline{I}^{n+1} d\Omega$

gather $\underline{f}^{int} \leftarrow \underline{f}_e^{int}$

if $n_{cyc} = 0$, compute Δt_G

$t_G \leftarrow t_G + \Delta t_G$

4. nodal integration loop, $N=1, N_{node}$

if $n_{cyc} = 0$, set Δt_N

if $t_N \leq t_{MAST}$

update $\underline{v}_N, \underline{u}_N, t_N \leftarrow t_N + \Delta t_N$

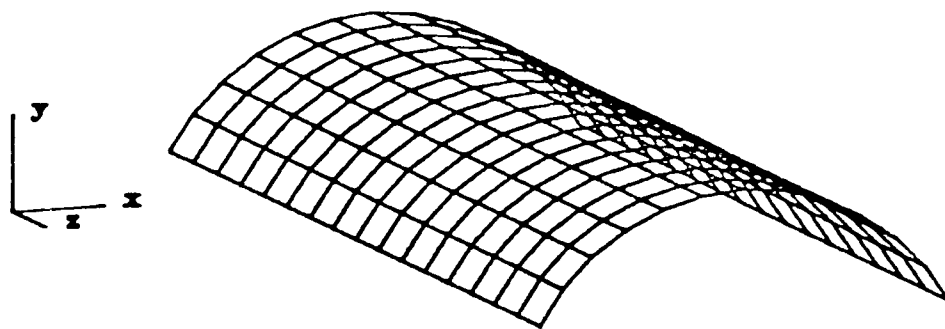
5. $n_{cyc} \leftarrow n_{cyc} + 1, n \leftarrow n + 1$; if $n_{cyc} = n_{cyc}^{max}$ then $n_{cyc} = 0$

SLIDE 7

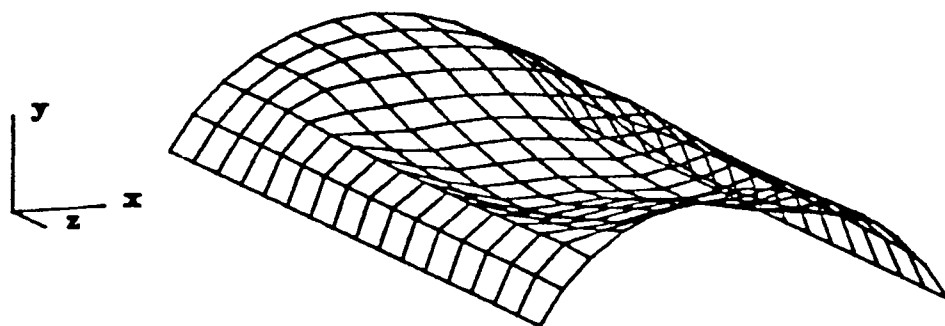
This slide shows the flow of computations for a structural dynamics program with mixed time explicit integration. The major difference from the standard flow is that it is necessary to set up a number of clocks:

1. the master time, which governs the evolution of the problem;
2. the element block times, which indicate the time to which each of the element blocks has been updated;
3. nodal times (for each node), which indicate to what time the node has been updated.

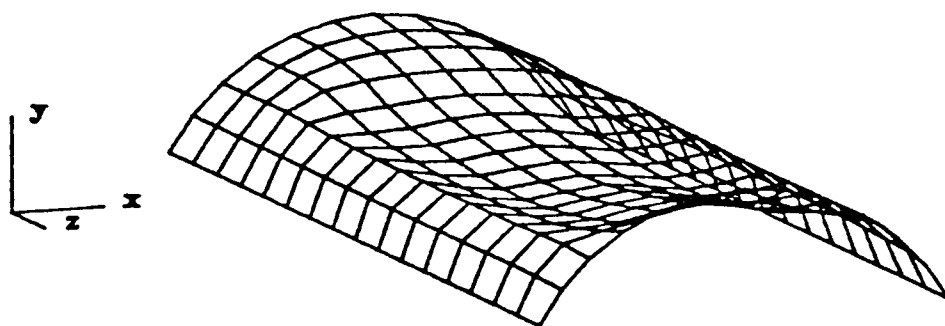
The structure of the algorithm is almost identical to a standard nonlinear structural dynamics program, except for the fact that element groups and nodes are only updated when their clocks are behind the master time.



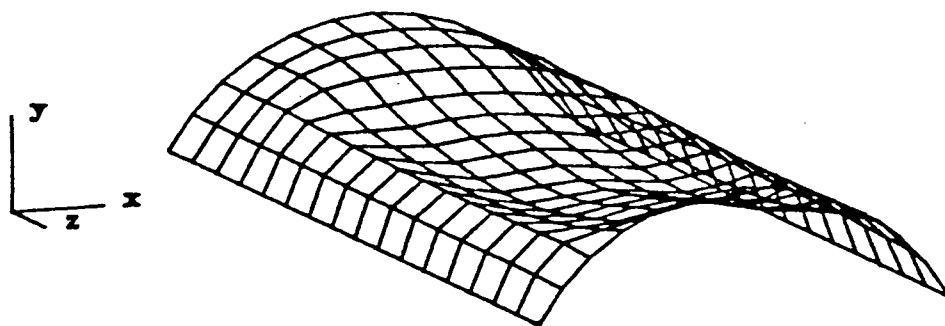
time =.378E-03



time =.600E-03



time =.100E-02



SLIDE 8

SLIDE 8

This is the first example we studied. It is an impulsively loaded cylindrical shell. The problem involves very large displacements and elastic-plastic response. It is solved with several meshes of quadrilateral shell elements. The shell elements are four-node elements with one-point quadrature and stabilization as described in "Explicit Algorithms for the Nonlinear Dynamics of Shells" (T. Belytschko, J. I. Lin, and C.-S. Tsay, Computer Methods in Applied Mechanics and Engineering, Vol. 42, 1984, pp. 225-251).

Table

Cylindrical Panel Timing Study (Seconds of CPU)

Program Version	Number of Processors	Mesh 1		Mesh 2		Mesh 3		
		96 elements 12 ele/blk	24 ele/blk	384 elements 32 ele/blk	48 ele/blk	1536 elements 32 ele/blk	192 ele/blk	
Original WHAMS	1	347	2658			20860		
	8	141	1072			8484		
WHAMS-VECPAR	1	110	669	638	626	4891		4700
	2	66 (83%)	394 (85%)		380 (82%)	2937 (83%)		
	3	55	330		321			
	4	45	270	270	266	1994		2001
	5	44	268		265	1792		
	6	44	242		264	1709		1961
	7	45	240		265	1615		
	8	36 (38%)	231 (36%)	232	230 (34%)	1590 (38%)		1722

SLIDE 9

This gives various timings for the previously described problems. The rows labeled "Original WHAMS" pertain to timings on a version of this program which was not designed for concurrency or vectorization. This version of the program was run both with and without the compiler optimization on the Alliant FX/8. Note that in going from one to eight processors and compiler optimization, a speed-up of a factor only slightly greater than 2 was obtained. In the reprogrammed version, which is labeled "VECPAR", an improvement of 3 in running time is obtained in going from the original version of WHAMS. This speed-up is almost entirely due to taking advantage of vectorization. The increases in speed in going from one to eight processors are shown below that. Note that the speed-up in going to two processors is usually about 1.7, which is about 85% efficiency, whereas the speed-up in going to eight processors is only about a factor of 3, which indicates an efficiency of about 35%. Note that as the size of the blocks is increased, computer running time decreases, which indicates the speed-up due to vectorization.

Table

Timings for Containment Vessel Problem
with and without Mixed Time Integration (in sec CPU)

Program Version	Number of Processors	Single Δt	Mixed Time Integration			
		43 ele/blk	8 ele/blk	11 ele/blk	22 ele/blk	43 ele/blk
Original WHAMS	1	1325				
Original WHAMS with Compiler Optimization	8	538				
WHAMS-VECPAR	1	275			168	183
	2	171			101	115
	3	145			83	97
	4	122			71	79
	5	122			64	78
	6	122			61	78
	7	123			61	72
	8	108		63	52	67

SLIDE 10

This problem, which is labeled a containment problem, contains elements of various sizes, so it can be used to examine the effects of mixed time integration. The table contains timings for runs with both a single time step and mixed time integration. As in the previous problem, there is a significant improvement in the vectorized version of the program: running time on a single processor diminishes from 1325 seconds to 275 seconds. As the number of processors is increased in the VECPAR version, the incremental efficiency of additional processors diminishes somewhat. Again, speed-up of about 1.7 is obtained in going to two processors, and a factor of about 2.7 is obtained in going to eight processors. The speed-up in mixed time integration depends on the size of the element block and is actually somewhat better for small element blocks. There are two reasons for the increase in time with increasing element block size.

1. With increasing element blocks, idle time increases.
2. Since the time steps are automatically allocated to each block, the average time step for the blocks decreases when they are larger.

In comparing the single time step version of the original WHAMS with the single time step version of WHAMS-VECPAR, we see an improvement of a factor of 5 in running time. Adding mixed time integration provides another factor of 2, so the current work has been able to yield a twentyfold improvement in running time in this problem.

CONCLUSIONS

- REPROGRAMMING IS NECESSARY TO TAKE ADVANTAGE OF VECTORIZATION AND CONCURRENCY -- IT HAS YIELDED 10-FOLD SPEED INCREASE IN ALLIANT FX/8
- MIXED TIME INTEGRATION HAS SPEEDED UP EXECUTION BY 2
- VECTORIZATION AND CONCURRENCY ARE DIFFICULT TO EXPLOIT SIMULTANEOUSLY

SUGGESTED WORK

- IMPLEMENTATION OF NEW MIXED TIME INTEGRATION WITH NONINTEGER RATIOS
- ITERATIVE IMPLICIT METHODS
- INCORPORATION IN TEST BED

SLIDE 11

We list our conclusions and suggestions for future work.

A major new development would be to extend the method for mixed time integration so that the ratios of adjacent time steps need not be integer values. This would yield further speed-ups in the technique. We are also interested in implementing iterative methods on vectorized, concurrent machines because they pose similar scheduling-allocation problems, but the payoffs are substantial.

PUBLICATIONS SUPPORTED BY GRANT

- P. Smolinski, T. Belytschko and M. Neal, "Multi-Time Step Integration Using Nodal Partitioning," to appear, International Journal for Numerical Methods in Engineering.
- T. Belytschko and N. Gilbertsen, "Concurrent and Vectorized Mixed Time, Explicit Nonlinear Structural Dynamics Algorithms," to appear, Proceedings of Symposium on Parallel Computations and Their Impact on Mechanics, ed. A. K. Noor, ASME, New York, 1987; also submitted to Computer Methods in Applied Mechanics and Engineering.
- T. Belytschko and M. Neal, "A New Mixed Time Integration Method for Structural Dynamics," in preparation.

PRESENTATIONS

- T. Belytschko, N. Gilbertsen and J. M. Kennedy, "Explicit Time Integration Finite Element Codes Adapted to Parallel Computers"
- - Invited paper presented at Session on Computer Codes and Methods, 9th International Conference on Structural Mechanics in Reactor Technology, Lausanne, Switzerland, August 17-21, 1987.
- T. Belytschko and N. Gilbertsen, "Concurrent and Vectorized Mixed Time, Explicit Nonlinear Structural Dynamics Algorithms"
- - Paper to be presented at Symposium on Parallel Computations and Their Impact on Mechanics, 1987 ASME Winter Annual Meeting, Boston, Massachusetts, December 13-18, 1987.